REMARKS/ARGUMENTS

At page 6, sections 4-6, of the aforementioned Office Action, the Examiner rejected claims 1, 6 and 9 under 35 U.S.C. § 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner alleged that certain elements of claims 1, 6 and 9 did not have sufficient antecedent basis. Accordingly, each of claims 1, 6 and 9 have been specifically amended to address the Examiner's concerns. Accordingly, withdrawal of such rejection is respectfully traversed.

At page 6, sections 7-8, the Examiner rejected claims 1, 2, 6 and 7 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,633,908 ("Leymann") in view of "well known in the art." Such rejection is respectfully traversed.

Each of independent claims 1, 6 and 9 now expressly states that software code is added to the computer application being monitored by the API of the present invention to assign a single general reference to characteristic transactional information associated with a computer application transaction under surveillance. In stark contrast, Leymann does so by way of a separate application invocation agent. The following passages from the Leymann patent underscore the differences between his invocation agent based method of operation and the Applicants' invention:

> The basic idea of the present invention is to instrument not the application components. The present invention contemplates instrumenting the invocation agent instead, which in turn is responsible for calling the application for execution.

Leymann at col. 7, line 66, through column 8, line 1 (emphasis added).

> The present invention relates to the area of systems management teaching means and a method for determining and managing application performance. Application Response Measurement (ARM) assumes that the managed application is a self-instrumented component. This requires invasive changes to existing applications or to add explicitly code to newly written applications. Due to this additional effort this will restrict the area of applicability of ARM. The basic idea of the present invention is to instrument not the application components. The present invention contemplates instrumenting the invocation agent instead, which in turn is responsible to call the application for execution. This solution provides application response measurement without any modification of the application being measured. It is the invocation agent that makes the appropriate ARM calls to furnish the instrumentation on behalf of the application.

Leymann at Abstract(emphasis added).

Leymann is thus diametrically opposed in architecture and operation to the system and method recited in Applicants' independent claims.

The Examiner paints the many specific limitations of Applicants' independent claims with a very broad brush in the form of a passage spanning column 2, line 44 through column 3, line 24 of Leymann. It appears, however, that the Examiner draws inspiration from a slightly larger segment of the Leymann text, *viz.*, column 3, line 41 through column 4, line 26, which is reproduced below in its entirety.

> There are different and incompatible ARM products in the market place. Without the present invention the application provider has to decide which of the systems management environments to adhere to, which in the worst case means that he has to furnish for all of them. The present invention allows one to make this decision on the application integration level. Moreover as the invocation agent has the information on which application it has to start, the present invention is flexible enough to allow one to make the decision, which ARM product to involve, on the basis of each individual application. Therefore in accordance with the present invention an application is (to a certain extent) decoupled from the specific ARM product.
>
> Additional advantages are accomplished in a preferred embodiment of the proposed invention in which the instrumentation means of the invocation agent is executed outside of the response measurement scope of the application being measured.
>
> By explicitly performing all invocation agent activities not directly relating to the application execution before starting or after terminating the response measurement, it is guaranteed that the measured data are precise and relate to the application execution and not to the processing of the invocation agent.
>
> Additional advantages are accomplished in a preferred embodiment of the proposed invention in which the instrumentation means further comprises application response measurement setup means for requesting the ARM to measure the response of the application instance and application response measurement

> termination means for requesting the ARM to terminate the response measurement.
>
> Through these two distinctive means the invocation agent is enabled to precisely control the "time window", in which the ARM will associate the response measurement data to the application. Such a feature allows the invocation agent to perform extra processing, which will not enter the response measurement data of the application. Thus it is guaranteed that the measured data are precise and relate to the application execution and not to the processing of the invocation agent.
>
> Additional advantages are accomplished in a preferred embodiment of the proposed invention in which the application response measurement setup means further identifies a transaction of the application instance to the ARM to be measured. The application response measurement termination means further identifies the transaction to the ARM for which the response measurement is to be terminated.
>
> The present invention makes maximal use of information available to the invocation agent. As the invocation "knows" which application/transaction it has to invoke it is also able to share this information with the ARM. The ARM is thus able to associate the measured data with the correct application/transaction.

Nowhere in the foregoing passage -- or anywhere else -- does Leymann expressly or impliedly disclose an application program interface that adds software code to a monitored computer application for assigning a single general reference to characteristic transactional information associated with a transaction to be executed by said computer application, as is required by each of Applicants' independent claims 1, 6 and 9. Applicants discuss this significant, novel and unobvious feature in the paragraph bridging pages 14 and 15 of Applicants' specification, which is reproduced below (with emphasis added).

> The API according to the present invention is placed strategically in a computer application to mark the beginning and end of processing (and any other significant events) at desired computer system components or processes, all of which are selected at the discretion of the user of the API. More specifically, API software code is added to the computer application which, when executed, assigns a single general reference to characteristic transactional information associated with a transaction event to be executed by the computer application. Additionally, the API includes an agent that marks the time at which the API software code is executed and tags that time with the business or other transactional information being currently processed by the computer application. Unlike conventional ARM APIs, the present API does not create or pass any data from one system component to the next (e.g., a timestamp or a unique API-generated handle, correlator or other identifier) beyond that of the business information ordinarily passed in processing a transaction. That is, the present invention recognizes that characteristic transactional information inherently associated with a given transaction, in and of itself, constitutes a readily identifiable electronic fingerprint or reference that is sufficient to enable identification and tracking of events processed by a computer application in executing the transaction as it flows through a computer system. For instance, characteristic business or other transactional information associated with a securities trade may include, inter alia, a Trade Identifier (or trade ID or trade reference, the identity of the party requesting the trade, the type of securities being traded, the number of securities being traded, the price of the securities, the date of the trade, whether the trade is a "buy" or a "sell", as well other trade-specific information. Thus, the aggregation of this characteristic transactional information represents a unique identifier that itself

> may be directly tracked throughout processing by a computer system, thereby eliminating the need for a new and different API-generated handle to be created, correlated and tracked at each transition from one computer system component to the next and for each computer application transaction conducted in executing the transaction.

With due respect, it is legal error to equate the notion that the invocation agent of the Leymann system "knows" which application/transaction it is to invoke, and is also able to share this information with the ARM such that the ARM is able to associate the measured data with the correct application/transaction (Leymann at column 4, lines 21-26), with Applicants claimed system wherein an API adds software code to the computer application which, when executed, assigns a single general reference to characteristic transactional information associated with a transaction event to be executed by the computer application. This specific beneficial concept is simply not taught by Leymann.

In the first three paragraphs on page 9 of the Office Action, the Examiner states (with emphasis added):

> However, Leymann's invention does not teach adding software code to said computer application.
>
> The limitations, "adding software code to said computer application", are well known in the art, for example, abstract, col. 7, line 51 - col., 8, line 15.
>
> It would have been obvious to one of ordinary skill in the art at the time the invention was made to include adding software code to said computer application with the teachings of Leymann in order to facilitate

> addition of software code to the application because the code would help enhance the functionality of the application. The application with enhanced functionality would provide support for monitoring information.

The Examiner's comments here are faulty for two reasons. First, the essence of the Leymann invention is to not add software code to the application under surveillance. To suggest that it would be obvious to modify the Leymann system to do so based a couple brief references to "well known" prior art -- when the Leymann patent strenuously advocates against doing so -- would not only be illogical but would also vitiate the central purpose of the Leymann invention. Second, neither of the passages in Leymann relied upon by the Examiner (the Abstract and column 7, line 51 through column 8, line 15) describe or imply using an API to add software code to a monitored computer application for assigning a single general reference to characteristic transactional information associated with a transaction event to be executed by the application.

Consequently, Leymann neither anticipates, suggests nor otherwise renders obvious the invention claimed by Applicants. Applicants therefore submit that the outstanding Section 103(a) rejection of claims 1, 2, 6 and 7 is improper and should be withdrawn.

At page 10, section 11, of the Office Action, the Examiner rejected claims 3, 8 and 9 under 35 U.S.C. § 103(a) as being unpatentable over Leymann in view of "well known in the art" and

U.S. Patent No. 6,108,700 (Maccabee et al., "Maccabee"). Such rejection is respectfully traversed.

Applicants' comments concerning Leymann and the so-called "well known" prior art are incorporated herein and reasserted in their entirety by reference thereto.

Notwithstanding what Maccabee may or may not disclose in relation to Applicants' claims 3 and 8, that patent teaches directly away from independent claims 1 and 6 from which they depend (as well as independent claim 9). More particularly, Maccabee proposes the creation of a transaction definition language called the ETE (End-to-End) Transaction Definition Language that specifies how to construct identifiable transactions from events and links. In an illustrated example, the ETE Transaction Definition Language provided in Maccabee requires the creation of twenty-one (21) lines of software code merely to define something as relatively simple as a Web commerce transaction. Merely contemplating all of the possible events and transactions that might be involved in a complex business transaction, particularly one whose execution involves the coordination of several business entities and computer systems, is itself a daunting task. Codifying these items complicates the task. That is, individually defining all of these events and transactions in software code in order to produce a complete set of transaction generation rules amounts to a potentially vast amount of preliminary preparation activity that must be performed before the monitoring system may be placed into operation.

The present invention requires no predefinition of events and thereby avoids the disadvantages of the Maccabee system. This beneficial feature is present in each of independent claims 1, 6 and 9. The pertinent passage from claim 1 is representative:

> without predefining events describing the potential stages of a transaction to be executed by said computer application, using an application program interface to add software code to said computer application for assigning a single general reference to characteristic transactional information associated with said transaction

Maccabee is therefore in direct conflict with an essential feature of each Applicants' independent claims. Accordingly, since Maccabee leads one directly away from the invention prescribed in those claims, it necessary also leads one directly away from the invention defined by their dependent claims, including claims 3 and 8. As a result, since no combination of the teachings of Leymann and Maccabee can produce Applicants invention as defined by independent claims 1 and 6 (and 9), the Leymann-Maccabee reference tandem likewise cannot render obvious dependent claims 3, 8 and 9. Withdrawal of the outstanding Section 103(a) rejection of claims 3, 8 and 9 is therefore respectfully requested.

At page 14, section 14, of the Office Action, the Examiner rejected claims 4, 5, 10, 11 and 12 under 35 U.S.C. § 103(a) as being unpatentable over Leymann and Maccabee in view of "Official Notice." Such rejection is respectfully traversed.

Applicants' comments concerning Leymann and Maccabee are incorporated herein and reasserted in their entirety by reference thereto.

In short, the teachings of Leymann and Maccabee cannot be combined in any way to so as to anticipate or render obvious independent claims 1, 6 and 9. As a matter of law, therefore, Leymann and Maccabee cannot be relied upon to reject the claims that depend therefrom, including claims 4, 5, 10, 11 and 12 -- notwithstanding the Examiner's invocation of "Official Notice" in connection with those claims (which "Official Notice" Applicants nevertheless do not concede). Even assuming, *arguendo*, the newly cited references to Lee et al., Sager et al., Klein et al., Schweitzer et al. Brede et al. and Paley et al. generally disclose the notions of "handling of latency information" and "calculating latency between components," they do not do so in the specific way or using the specific formula posited and claimed by Applicants. Hence, "Official Notice" of those patents does not cure their deficiencies or those of the underlying Leymann-Maccabee reference combination vis-à-vis Applicants' claims.

Accordingly, withdrawal of the outstanding Section 103(a) rejection of claims 4, 5, 10, 11 and 12 is respectfully requested.

In view of the foregoing, the instant application is believed to be in condition for allowance and, therefore, early issuance thereof is earnestly solicited.

If the Examiner believes that a telephone interview would be beneficial to advance prosecution of the present application, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Date: November 7, 2005

John F. Letchford
Registration No. 33,328

ARCHER & GREINER, P.C.
One Centennial Square
Haddonfield, NJ 08033
Tel.: (856) 354-3013
Fax: (856) 795-0574
Email: jletchford@archerlaw.com

2039826v1